ACTIVE: Activity Concept Transitions in Video Event Classification

Chen Sun and Ram Nevatia
University of Southern California, Institute for Robotics and Intelligent Systems
Los Angeles, CA 90089, USA
{chensun|nevatia}@usc.edu

Abstract

The goal of high level event classification from videos is to assign a single, high level event label to each query video. Traditional approaches represent each video as a set of low level features and encode it into a fixed length feature vector (e.g. *Bag-of-Words), which leave a big gap between low level visual features and high level events. Our paper tries to address this problem by exploiting activity concept transitions in video events (ACTIVE). A video is treated as a sequence of short clips, all of which are observations corresponding to latent activity concept variables in a Hidden Markov Model (HMM). We propose to apply Fisher Kernel techniques so that the concept transitions over time can be encoded into a compact and fixed length feature vector very efficiently. Our approach can utilize concept annotations from independent datasets, and works well even with a very small number of training samples. Experiments on the challenging NIST TRECVID Multimedia Event Detection (MED) dataset shows our approach performs favorably over the state-of-the-art.*

1. Introduction

Video event classification is an important computer vision problem needed for many tasks including automatic tagging and content based retrieval. Early work deals with well-defined atomic actions such as *walking*, *kissing* and *hand shake* [17] [10]. The videos are usually short clips taken in constraint environment. Compared with these, high level event classification for web videos focuses on classifying complex events (*e.g. wedding ceremony*, *feeding an animal*) from large number of videos in the wild. It poses several key challenges: First, high level events usually involve multiple human-object interactions. There is a hiearchy where events can be decomposed into several mid-level activities (*e.g. dancing*), each of which consists of atomic actions (*e.g. walking*). We refer the latter two as activity concepts. Second, the videos are captured by amateurs, with different video qualities, irregular camera motions, shot changes, and huge intra-class variation. Finally, the datasets usually contain large number of videos.

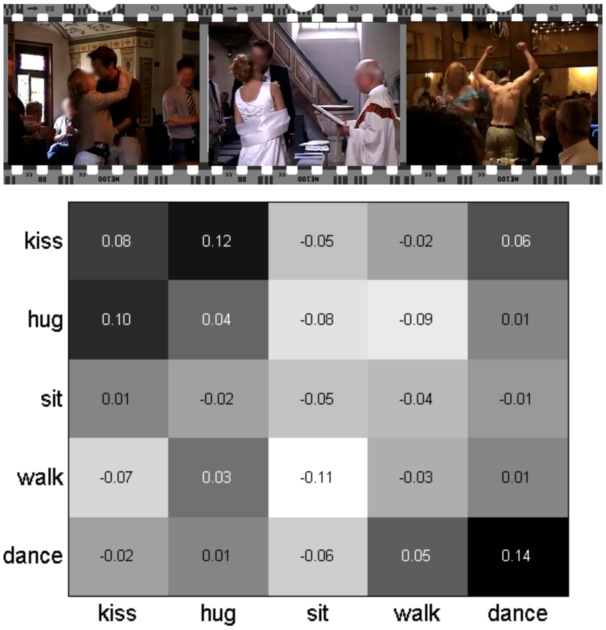


Figure 1. Illustration of our approach. (**Top**) A video from *Wedding ceremony* event is separated into a sequence of clips, each of which corresponds to an activity concept like *kissing* and *dancing*. (**Bottom**) Each dimension in our representation corresponds to an activity concept transition. A positive value indicates the transition is more likely to happen than parameterized by a Hidden Markov Model.

Facing these challenges, the current state-of-the-art takes a simple approach yet achieves impressive results. It follows the Bag-of-Words (BoW) scheme with the following three steps: low-level feature extraction, fixed-length feature vector encoding and classification. The features may include local image descriptors [13], motion descriptors [8] [9] [23] or audio descriptors. Effective as it is, there are some limitations. First, unlike lower level activity concepts which have relatively discriminative motion or visual patterns, most high level events consist of complex human object interaction and various scene backgrounds, which pose difficulty for the existing low level frameworks. Second, low level features are usually encoded into a histogram over the entire video, dropping useful temporal information. Finally, many activity concepts have pairwise relationships in temporal domain. These relationships provide useful clues for classifying high level events. For example, *kissing* is usually followed by *hugging* in a *wedding ceremony* event, as shown in Figure 1.

To overcome these limitations, we propose to encode activity concept transitions with Fisher kernel techniques [7]. The basic idea is to extract statistics information from some generative model for classification in a discriminative approach. It represents a set of data by its derivative of log-likelihood function over the set of model parameters, which is used as input for a discriminative classifier like a Support Vector Machine (SVM). Intuitively, it measures the difference of the incoming data from an underlying model. Here we use Hidden Markov Model (HMM) as the underlying generative model. In this model, a video event is a sequence of activity concepts. A new concept is generated with certain probabilities based on the previous concept. An observation is a low level feature vector from a sub-clip and generated based on the concepts. By using this model, we bridge low level features and high level events via activity concepts, and utilize the temporal relationships of activity concepts explicitly. We call the vector produced by applying the Fisher kernel to an HMM to be an HMM Fisher Vector (HMMFV). Our approach has the following features:

No maximum a posteriori (MAP) inference needed. HMM in traditional generative framework requires MAP inference to find out the concept assignments over time with the highest probability. A separate model for each new event is needed. Instead, HMMFV only uses HMM for the purpose of vector generation, and can utilize a single general model for all videos.

Efficient for large-scale usage. HMMFV is a compact and fixed-length feature vector. It can be used directly in most classification and retrieval frameworks using low-level features. HMMFV has a closed form representation and can be computed by dynamic programming very efficiently.

Robust with limited training data. Our activity concept classifiers are pre-learned and offer a good abstraction of low level features. This makes it possible to learn robust event classifiers with very limited training data.

Our approach can utilize both mid-level and atomic activity concepts, even when they do not occur in high level events directly, or are collected from a different domain. In this case, activity concepts can be seen as groups of low level features such that each of them can provide useful statistics. Besides, each dimension of HMMFV corresponds to a concept transition. As illustrated in Figure 1, dimensions with high values correspond to highly possible transitions, and can be used to describe the video.

The key contributions of this paper are threefold: First, we propose to encode activity concept transitions with Fisher Vectors for high level event classification and description. Second, we derive HMMFV over the transition parameters, which has a closed form representation and can be computed efficiently. Third, we provide detailed experiments and analysis showing our approach enjoys better performance over state-of-the-art in real world settings.

2. Related Work

Low-level features are widely used in high level video event classification. There are static features like SIFT [13], and motion features like STIP [9] and Dense Trajectories (DT) [23]. Feature descriptors can be used either densely or on the interest points only. Though sparse features are more compact in size, it is shown that dense features have better performance in various datasets [24]. For high level video event classification, [20] evaluates different types of low-level visual features, and shows a late fusion of these features can improve performance.

The idea of using concepts has been adopted in image [22] [11] [3] and video classification [12] under different names. A set of object concept classifiers called *classemes* is used for image classification and novel class detection tasks in [22]. Meanwhile, Object Bank [11] applies object filters over different locations instead of using whole image based classemes. Our activity concepts, on the other hand, are detected on fixed length short video clips. Like [22] and [11], our framework doesn't require the concepts classifiers to be perfect or directly related to target domain.

Recently, [5] models the activity concepts as latent variables with pairwise correlations, and applies latent SVM for classification. Unlike our approach, their activity concepts have single responses for the entire video and cannot model the evolution of concepts over time. Among the frameworks which also exploit temporal structure, [21] uses explicit-duration HMM where both concepts and concept durations are hidden variables, and [4] uses a generative temporal model by estimating the distribution of relative concept spacings. Our approach is different from them as our model with activity concepts is not used for classification directly.

Fisher Vector is proposed by [7]. It is introduced to image classification task in [15], where the underlying model is Gaussian Mixtures (GMM). A similar approach is used in high level video event classification and shows promising results [19]. The Fisher Vector of HMM has been derived for the emission probability parameters in [6]. In this paper, we give a brief derivation of HMMFV over transition probability parameters.

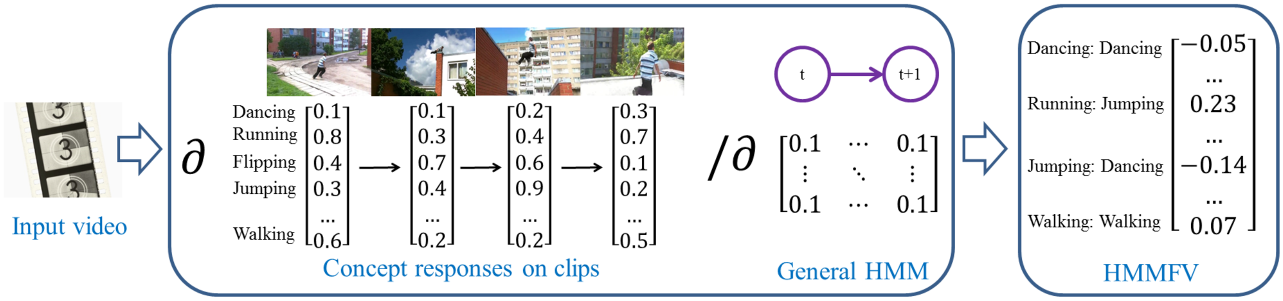


Figure 2. Illustration of our HMMFV generation process. An input video is separated into fixed length clips, each has a vector of activity concept responses. HMMFV is computed by taking the partial derivatives of the video's log-likelihood in a general HMM over the transition parameters. Each dimension in HMMFV corresponds to a concept transition.

3. Video Representation

In this section, we describe how we represent videos with activity concepts, as well as how to get the representation. To avoid confusion, we first define the term **events** and **activity concepts** used throughout the paper.

- An *activity concept* is an atomic action or an activity containing simple interactions among objects over a short period of time. (*e.g*. less than 10 seconds)
- An *event* is a complex activity consisting of several *activity concepts* over a relatively long period of time. (*e.g*. 30 seconds or higher)

The activity concepts we use are predefined and trained under supervision. Activity concept classifiers are built from low-level visual features. All techniques for event classification with low level features can be used, resulting a single fixed-length descriptor $\mathbf{x}$ for each video clip.

We then train 1-vs-rest classifier ϕ_c for each activity concept c. Since $\mathbf{x}$ is usually high dimensional, we use linear SVM to save training and prediction time. The output of $\phi_c(\mathbf{x})$ is defined as the probability output returned by LIBLINEAR [2] for $\mathbf{x}$ under Logistic Regression Model.

After the concept classifiers $[\phi_1\ \phi_2\ ...\ \phi_K]$ are obtained, we scan the video with fixed-length sliding windows and represent each video by a T by K matrix $M = [\phi_{t,k}]$, where T is the number of sliding windows, K is the number of activity concepts and $\phi_{t,k}$ is the classifier response of the k-th activity concept for the t-th sliding window.

4. HMM Fisher Vector

In this section, we introduce how we model and encode the transitions of activity concepts in videos. Figure 2 gives an illustration of the whole process.

4.1. Our Model

We use HMM to model a video event with activity concept transitions over time. There are K states, each corresponds to an activity concept. Every two concepts i, j have a transition probability $P(C_j|C_i)$ from concept i to j. Each observation is a feature vector $\mathbf{x}$ extracted from a sliding window.

Since we are working with a generative model, the emission probability of $\mathbf{x}$ given concept C_i is derived from

$$P(\mathbf{x}|C_i) \sim \frac{\phi_i(\mathbf{x})}{P(C_i)}$$

where $\phi_i(\mathbf{x})$ is the activity concept classifier output and $P(C_i)$ is the prior probability of concept i. Here we assume uniform prior for all observations.

To make the derivation clearer, we define the following notations:

- $\boldsymbol{\pi_i}$ is the prior probability of concept i.
- $\boldsymbol{\tau_{i|j}}$ is the transition probability from concept j to concept i.
- $\boldsymbol{\theta_{x|i}}$ is the emission probability of $\mathbf{x}$ given concept i.

4.2. HMMFV Formulation

The idea of Fisher kernel is first proposed in [7], the goal is to get the *sufficient statistics* of a generative model, and use them as kernel functions in discriminative classifiers such as SVM.

Denote $P(X|\theta)$ is the probability of X given parameters θ for some generative model, Fisher kernel is defined as

$$U_X = \nabla_\theta \log P(X|\theta) \tag{1}$$

$$K(X_i, X_j) = U_{X_i}^T I^{-1} U_{X_j} \tag{2}$$

where I is the Fisher information matrix.

If we ignore I by setting it to identity, U_X can be seen as a feature vector in linear kernel. In this paper, we use U_X as a feature vector and name it Fisher Vector for some generative model.

As the emission probabilities are derived from activity concept classifiers, we only use partial derivatives over transition probability parameters $\tau_{i|j}$ to derive the Fisher Vector U_X. Besides, we decide not to include event label variable since it makes the dimension of Fisher Vector growing linearly with the number of events, and requires recomputation of U_X every time there is a new event.

The log-likelihood of HMM is given by

$$\log P(X|\theta, \tau) = \log \sum_{s_1,\ldots,s_n} \prod_{i=1}^{T} \theta_{x_i|s_i} \tau_{s_i|s_{i-1}} \tag{3}$$

where $s_1, \ldots, s_n$ are enumerating all the possible activity concepts. To simplify notation, let $\tau_{s_1|s_0} = \pi_{s_1}$.

By taking the partial derivative of the log-likelihood function over $\tau_{i|j}$, we have

$$\frac{\partial}{\partial \tau_{i|j}} \log P(X|\theta, \tau) = \sum_t \left[\frac{\xi_t(i,j)}{\tau_{i|j}} - \gamma_{t-1}(j) \right] \tag{4}$$

where

$$\xi_t(i,j) = P(s_t = i, s_{t-1} = j | X, \theta, \tau)$$
$$\gamma_{t-1}(j) = P(s_{t-1} = j | X, \theta, \tau)$$

Denote

$$\alpha_t(i) = P(x_1, \ldots, x_t, s_t = i | \theta, \tau)$$
$$\beta_t(i) = P(x_{t+1}, \ldots, x_n | s_t = i, \theta, \tau)$$
$$FV(i,j) = \frac{\partial}{\partial \tau_{i|j}} \log P(X|\theta, \tau)$$

We have

$$FV(i,j) \sim \sum_t \alpha_{t-1}(j) \left[\theta_{x_t|i} \beta_t(i) - \beta_{t-1}(j) \right] \tag{5}$$

The vector is then normalized to make its L^2-norm as 1.

4.3. Parameter Learning

We use only one general model for HMMFV. When there are new event classes, instead of relearning HMM parameters, we can still use the same model without changing existing HMMFVs and only update the discriminative classifiers.

Here we use a very simple method to learn the model. First, we randomly select activity concept responses from neighboring sliding windows of all events. Model parameters are computed as

$$\tau_{i|j} \sim \sum_k \phi_i(\mathbf{x}^{k,1}) \phi_j(\mathbf{x}^{k,0}) \tag{6}$$

$$\pi_i \sim \sum_k \phi_i(\mathbf{x}^{k,0}) \tag{7}$$

where $\mathbf{x}^{k,0}$ and $\mathbf{x}^{k,1}$ are neighboring observations, and k is over all the samples.

Then we normalize the parameters to make them valid distributions.

4.4. Discussions

HMMFV can be computed very efficiently. The α's and β's required for all $FV(i,j)$ can be computed via standard dynamic programming [16]. The dimension of HMMFV is K^2, where K is the number of activity concepts.

Intuitively, by looking into Equation 4, HMMFV accumulates the difference between the actual expectation of concept transition and the model's prediction based on the previous concept. If the observations fit the model perfectly, the difference is zero. As we are using a single general model, background information is thus suppressed. This is especially useful for high level event classification since the videos often contain irrelevant information.

By taking derivatives of model parameters, HMMFV preserves the *sufficient statistics* of HMM. Consider a birthday party event, in which *people singing* and *people dancing* are often followed by each other, $FV(\textit{dancing}, \textit{singing})$ and $FV(\textit{singing}, \textit{dancing})$ should both have high positive energy, indicating their transition probabilities are underestimated in the general model. Similarly, $FV(\textit{washing}, \textit{sewing})$ should have high negative energy, indicating their transition probability is overestimated. Based on this property, we can describe a video using activity concept transitions with high positive values in HMMFV.

5. Experiments

In this section, we describe the dataset we used and the experiment settings. Then we compare our approach with baseline, and study the influence of activity concept selection. Finally we give performance comparison with two state-of-the-art systems, with abundant and few training samples.

5.1. Dataset

We used TRECVID MED 11 Event Kit data [1] (EventKit) for evaluation. This dataset contains 2,062 diverse, user-generated videos vary in length, quality and resolution. There are 15 event classes: 1. *Attempting a board trick*, 2. *feeding an animal*, 3. *landing a fish*, 4. *wedding ceremony*, 5. *working on a woodworking project*, 6. *birthday party*, 7. *changing a vehicle tire*, 8. *flash mob gathering*, 9. *getting a vehicle unstuck*, 10. *grooming an animal*, 11. *making a sandwich*, 12. *parade*, 13. *parkour*, 14. *repairing an appliance* and 15. *working on a sewing project*.

For the purpose of training activity concept classifiers, we used two datasets. We got 60 activity concept annotations used in [5] by communicating with the authors. The concepts were annotated on the EventKit and highly related to high level events. We call these concepts *Same Domain Concepts*, some of the concept names are shown in Table 1.

Another dataset we used for training concepts is the UCF 101 [18] dataset. It has 13,320 videos from 101 categories. Most of the videos in UCF 101 are less than 10 seconds long. The categories range from playing musical instruments to doing sports, most of which are not related to the events in EventKit directly. We call them *Cross Domain Concepts*.

5.2. Experimental Setup

To compare our framework with [5], we followed their protocol and randomly selected 70% videos from EventKit for training and 30% for testing. All the videos were resized to have 320 pixels in width. We set the size of sliding windows as 100 frames, with 50-frame overlap.

Dense Trajectory (DT) feature [23] was used as low level feature for activity concept classification. DT tracks points densely and describes each tracklet with its shape and the HoG, HoF, MBH features around the tracklet. We used the implementation provided by the authors [1], and set sampling stride to 10.

Recent results in image and video classification [15] [19] show that encoding low-level features with Gaussian Mixture Model (GMM) and Fisher kernel gives better performance than with BoW histograms. Suppose the low-level feature has D dimensions and the number of cluster centers for GMM is K, the resulting vector has $O(KD)$ dimensions. We used this encoding scheme and projected DT feature vectors to $D = 128$ with PCA, the number of clusters is 64. Since *same domain concepts* were annotated on all EventKit videos, we used only the annotations in training partition to train the activity concept classifiers. For *cross domain concepts*, we used all videos in UCF 101 as training data.

Max pooling (Max) was selected as the baseline, it represents a video by the maximum activity concept responses. Temporal information is dropped. Suppose the concept responses from the t-th sliding window is $[\phi_1^t\ \phi_2^t\ ...\ \phi_K^t]$, max pooling is defined as $[v_1\ v_2\ ...\ v_K]$ where

$$v_i = \max_t(\phi_i^t) \tag{8}$$

The vector is normalized to make its L^2-norm as 1, and used to build discriminative classifiers.

We used SVM classifier with RBF kernel for both HMMFV and Max, the parameters were selected by a 5-fold cross validation on training partition. Weighted average [14] was used to fuse results from different modalities.

All results were evaluated using average precision (AP).

[1] http://lear.inrialpes.fr/people/wang/dense_trajectories

Concept	AP	Concept	AP
Person running	0.059	Person dancing	0.146
Vehicle moving	0.330	Person marching	0.903
Person drilling	0.022	Person walking	0.068
Person kissing	0.147	Person flipping	0.154
Wheel Rotating	0.031	Person hammering	0.136
Animal approaching	0.015	Person carving	0.366
Hands visible	0.193	People dancing	0.146
Open door	0.038	Person singing	0.295
Taking pictures	0.038	Animal eating	0.204
Person cutting	0.017	Person sewing	0.185

Table 1. Average precision for same domain concepts

	Same Domain		Cross Domain	
Event ID	Max	HMMFV	Max	HMMFV
1	0.846	**0.857**	0.772	**0.806**
2	0.272	**0.398**	0.413	**0.458**
3	0.708	**0.767**	0.698	**0.748**
4	0.640	**0.782**	0.664	**0.717**
5	**0.525**	0.507	0.392	**0.646**
6	0.611	**0.753**	0.791	**0.831**
7	0.393	**0.492**	0.249	**0.355**
8	0.660	**0.745**	0.857	**0.864**
9	0.635	**0.730**	0.635	**0.687**
10	0.498	**0.539**	0.585	**0.606**
11	0.252	**0.386**	0.386	**0.436**
12	0.645	**0.761**	0.706	**0.741**
13	0.528	**0.863**	0.721	**0.818**
14	0.344	**0.596**	**0.600**	0.596
15	0.381	**0.545**	0.384	**0.545**
mean AP	0.529	**0.648**	0.590	**0.657**

Table 2. Average precision comparison for event classification with same domain concepts and cross domain concepts, bold numbers correspond to the higher performance in their groups

5.3. Same Domain Concepts

We first evaluate our framework with same domain concepts. Table 1 shows the performance of our concept classifiers, the parameters were selected by 5-fold cross validation. 20 of 60 concepts are randomly selected due to space limitation.

According to the second and third columns of Table 2, HMMFV achieves better performance in 14 of the 15 events. This result validates that by encoding concept transitions, HMMFV preserves more information and has more discriminative power.

Max pooling achieves better performance in *woodworking* event, this may happen when some concept classifiers have strong performance and are highly correlated to a single event (*e.g. person carving*).

In general, HMMFV achieves 11.9% higher performance over the baseline.

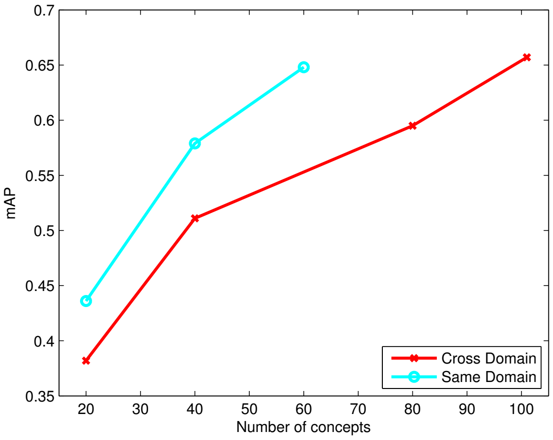


Figure 3. Mean average precisions with different number of concepts. The red line shows a randomly selected subset of cross domain concepts from 20 to 101. The cyan line illustrates a randomly selected subset of same domain concepts from 20 to 60.

5.4. Cross Domain Concepts

Although same domain concepts can provide semantic meanings for the videos, the annotations can be expensive and time consuming to obtain and adding new event classes can become cumbersome. Hence, we also studied the effect of using cross domain concepts.

Interestingly, the fourth and fifth columns of Table 2 show that even if the activity concepts are not related to events directly, our framework still achieves comparable performance. It is quite likely that the concept classifiers capture some inherent appearance and motion information, so that they can still be used to provide discriminative information for event classification. HMMFV still achieves better performance than max pooling, which indicates that temporal information is also useful when the activity concepts are from a different domain.

To study the influence of domain relevance, we randomly selected 20, 40 and 60 concepts from same domain concepts, and 20, 40 and 80 concepts from cross domain concepts for HMMFV. The mean AP performance is plotted in Figure 3. According to the figure, the performance increases with more concepts. Same domain concepts can easily outperform cross domain concepts given same number of concepts and reach the same level of performance with only 60 concepts, compared with 101 from cross domain concepts. Besides, as shown in Table 3, if we combine the two sets of results by late fusion, mean AP can be further improved by 4%, which indicates that HMMFV obtained from same and cross domain concepts have complementary performance.

Concepts	Same Domain	Cross Domain	Both
mean AP	0.648	0.657	**0.693**

Table 3. Mean average precisions when using same domain concepts, cross domain concepts and both for generating HMMFV

Event ID	Joint [5]	LL[15]	HMMFV	HMMFV+LL
1	0.757	0.813	**0.885**	0.882
2	**0.565**	0.363	0.468	0.461
3	0.722	0.743	**0.796**	0.789
4	0.675	**0.829**	0.771	0.811
5	**0.653**	0.496	0.604	0.623
6	0.782	0.736	0.811	**0.814**
7	0.477	**0.541**	0.482	0.518
8	**0.919**	0.868	0.873	0.877
9	0.691	0.769	0.756	**0.772**
10	0.510	0.579	0.617	**0.634**
11	0.419	0.515	0.476	**0.524**
12	0.724	0.720	0.770	**0.770**
13	0.664	0.792	0.886	**0.890**
14	**0.782**	0.661	0.619	0.634
15	0.575	0.608	0.575	**0.621**
mean AP	0.661	0.669	0.693	**0.708**

Table 4. Average precision comparison with state-of-the-art. Joint refers to the joint modeling of activity concepts, as described in [5], LL refers to the low level approach as described in [15]

5.5. Comparison with State-of-the-Art

In this section, we compare our framework with two state-of-the-art approaches. [5] is an activity concept based system. It models the joint occurrence of concepts without considering temporal information. We used their provided numbers and followed the same data partitioning method. We also compare our framework with a low level based approach: we implemented the Fisher kernel with visual vocabulary method in [15] but used only the components corresponding to μ. We used the same low level features (DT) for building our concept classifiers. We used both same domain and cross domain concepts in our framework. An event-by-event comparison is shown in Table 4.

Our framework outperforms the joint modeling of activity concepts approach in 11 of the 15 events. Moreover, we used only a single type of low level feature, and did not fuse the event classification results obtained from low level features directly. Compared with the low-level approach, our framework is better in 9 of the 15 events. Our framework achieves the best performance when used alone.

Besides, if we fuse the low level results with our framework, the overall performance can further increase to 70.8%, a 4% improvement from the two previous systems.

These comparisons indicate that encoding concept transitions provides useful information for event classification. Even though the joint modeling approach does consider

Method	mean AP
LL	0.421
Max (Same Domain)	0.456
HMMFV (Same Domain)	0.554
Max (Cross Domain)	0.432
HMMFV (Cross Domain)	0.470
HMMFV (Both)	**0.562**

Table 5. Average precision comparison with 10 training samples for each event

pairwise relationship of activity concepts, temporal information is not preserved.

5.6. Classification with Limited Training Samples

In real world retrieval problems, it is desirable to let users provide just a few video samples of some event they want before the system can build a reasonably good event classifier.

In this section, we studied the case when only 10 positive samples per event are provided for training. The training videos were randomly selected from the original training partition, and we used the previous concept classifiers since they didn't include test information.

As shown in Table 5, when the number of training samples is limited, the performance of low level approach decreases more significantly than activity concept based HMMFV, their relative mean AP difference is 14.1%. One possible explanation is that by using activity concepts, our framework has a better level of abstraction, which can be captured by discriminative classifiers even with a few training samples. Another interesting observation is that, when the number of training sample is limited, the performance of same domain concepts is 8.4% higher than the performance of cross domain concepts. This is understandable since some same domain concepts are highly correlated with high level events (*e.g. kissing* for *wedding ceremony*), they can help preserve highly discriminative information if their classifiers are strong.

Again, HMMFV outperforms the max pooling baseline.

5.7. Event Description with Concept Transitions

Finally, we show how to describe high level events with concept transitions.

In Section 4.4, we showed that $FV(i, j)$ has high positive energy if the transition probability from concept j to i is high, and is underestimated by the general model. A direct application is to sort the HMMFV values in descending order and use activity concept transitions with largest values to describe the video. Compared with the description in [5], our method returns not only the activity concepts, but also the transition patterns over time.

We show the event level descriptions in Figure 4, the HMMFVs were obtained by averaging over all HMMFVs from test videos of a single event.

Most of the descriptions are highly semantically meaningful. For example, in a *parkour* event, the top three concept transitions are: *jumping to jumping*, *flipping to jumping* and *dancing to jumping*. Some descriptions are not exact but also informative, like *spreading cream to hands visible* in a *making a sandwich* event.

6. Conclusion

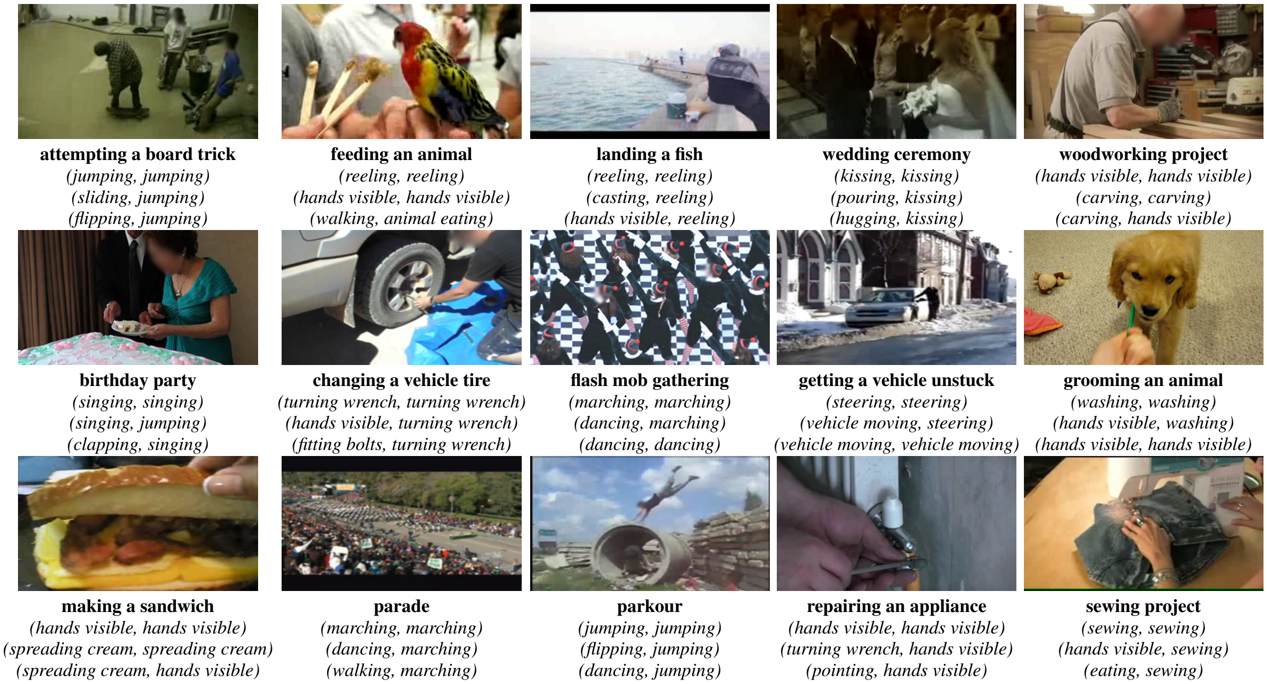
This paper addresses high level event classification problem by encoding the activity concept transitions over time. We chose HMM as the underlying model and applied Fisher kernel technique to obtain a fixed length description for the model. Our method is fast to compute and easy to use in existing frameworks. It can also be used to describe videos with activity concept transitions. Experimental results show that our approach achieves better results compared with state-of-the-art concept based framework and low level framework. Moreover, when the number of training samples is limited, our approach can still work reasonably well. Our system can utilize different types of activity concepts, we recommend same domain concepts for video description and compact HMMFV generation when they are available, and cross domain concepts to reduce the need for event specific concept annotations.

7. Acknowledgement

This work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract number D11PC0067. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes nonwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government. Computation for the work described in this paper was supported by the University of Southern California Center for High-Performance Computing and Communications (hpcc.usc.edu). We thank Dr. Mubarak Shah of University of Central Florida and Dr. Hui Cheng of SRI international Sarnoff for sharing the concept annotations. We also thank Boqing Gong, Song Cao and Chenhao Tan for helpful discussions.

References

[1] http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.11.org.html. 4



attempting a board trick
(jumping, jumping)
(sliding, jumping)
(flipping, jumping)

feeding an animal
(reeling, reeling)
(hands visible, hands visible)
(walking, animal eating)

landing a fish
(reeling, reeling)
(casting, reeling)
(hands visible, reeling)

wedding ceremony
(kissing, kissing)
(pouring, kissing)
(hugging, kissing)

woodworking project
(hands visible, hands visible)
(carving, carving)
(carving, hands visible)

birthday party
(singing, singing)
(singing, jumping)
(clapping, singing)

changing a vehicle tire
(turning wrench, turning wrench)
(hands visible, turning wrench)
(fitting bolts, turning wrench)

flash mob gathering
(marching, marching)
(dancing, marching)
(dancing, dancing)

getting a vehicle unstuck
(steering, steering)
(vehicle moving, steering)
(vehicle moving, vehicle moving)

grooming an animal
(washing, washing)
(hands visible, washing)
(hands visible, hands visible)

making a sandwich
(hands visible, hands visible)
(spreading cream, spreading cream)
(spreading cream, hands visible)

parade
(marching, marching)
(dancing, marching)
(walking, marching)

parkour
(jumping, jumping)
(flipping, jumping)
(dancing, jumping)

repairing an appliance
(hands visible, hands visible)
(turning wrench, hands visible)
(pointing, hands visible)

sewing project
(sewing, sewing)
(hands visible, sewing)
(eating, sewing)

Figure 4. High level events and their top rated descriptions based on activity concept transitions. Concept transitions with top 3 highest responses are listed under each event.

[2] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 2008. 3

[3] A. Farhadi, I. Endres, D. Hoiem, and D. A. Forsyth. Describing objects by their attributes. In *CVPR*, 2009. 2

[4] A. Gaidon, Z. Harchaoui, and C. Schmid. Actom sequence models for efficient action detection. In *CVPR*, 2011. 2

[5] H. Izadinia and M. Shah. Recognizing complex events using large margin joint low-level event model. In *ECCV*, 2012. 2, 4, 5, 6, 7

[6] T. Jaakkola, M. Diekhans, and D. Haussler. A discriminative framework for detecting remote protein homologies. *Journal of Computational Biology*, 7(1-2), 2000. 2

[7] T. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In *NIPS*, 1999. 2, 3

[8] A. Kläser, M. Marszalek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC*, 2008. 1

[9] I. Laptev. On space-time interest points. *IJCV*, 64(2-3):107–123, 2005. 1, 2

[10] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008. 1

[11] L.-J. Li, H. Su, E. P. Xing, and F.-F. Li. Object bank: A high-level image representation for scene classification & semantic feature sparsification. In *NIPS*, 2010. 2

[12] J. Liu, B. Kuipers, and S. Savarese. Recognizing human actions by attributes. In *CVPR*, 2011. 2

[13] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 1, 2

[14] P. Natarajan, S. Vitaladevuni, U. Park, S. Wu, V. Manohar, X. Zhuang, S. Tsakalidis, R. Prasad, and P. Natarajan. Multimodel feature fusion for robust event detection in web videos. In *CVPR*, 2012. 5

[15] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In *CVPR*, 2007. 2, 5, 6

[16] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, 1989. 4

[17] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In *ICPR*, 2004. 1

[18] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *CRCV-TR-12-01*. 5

[19] C. Sun and R. Nevatia. Large-scale web video event classification by use of fisher vectors. In *WACV*, 2013. 2, 5

[20] A. Tamrakar, S. Ali, Q. Yu, J. Liu, O. Javed, A. Divakaran, H. Cheng, and H. S. Sawhney. Evaluation of low-level features and their combinations for complex event detection in open source videos. In *CVPR*, 2012. 2

[21] K. Tang, F.-F. Li, and D. Koller. Learning latent temporal structure for complex event detection. In *CVPR*, 2012. 2

[22] L. Torresani, M. Szummer, and A. W. Fitzgibbon. Efficient object category recognition using classemes. In *ECCV*, 2010. 2

[23] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *CVPR*, 2011. 1, 2, 5

[24] H. Wang, M. M. Ullah, A. Kläser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *BMVC*, 2009. 2